CINVESTAV TAMAULIPAS
SISTEMAS DISTRIBUIDOS
PROF. DR. VICTOR SOSA SOSA
REPORTE: INSTALCIÓN DE AXIS DESDE CERO
DANIEL LÓPEZ ESCOGIDO

Tomcat es un contenedor de servlets (es lo que se necesita para ejecutar JSP y Servlets) creado por la fundación Apache dentro del proyecto Jakarta. Aunque se puede utilizar como servidor web no está tan optimizado como el servidor web de la misma fundación, Apache. En este reporte se compilarán los paquetes a partir del código fuente, en lugar de utilizar precompilados, por lo que debería poder seguirse facilmente utilizando cualquier distribución.

Instalación del JDK

Lo primero que necesitamos para desarrollar en Java es instalar el kit de desarrollo (JDK) que podemos descargar desde la web de Sun. Seguimos la siguiente secuencia de comandos:

```
su
```
Introducir contraseña de root
```
cp -p jdk-1_5_0_06-linux-i586.bin /usr/local
cd /usr/local
chmod +x jdk-1_5_0_06-linux-i586.bin
./jdk-1_5_0_06-linux-i586.bin
rm jdk-1_5_0_06-linux-i586.bin
```

Esto extraerá el contenido del archivo en una nueva carpeta jdk1.5.0_06 en /usr/local. Ahora basta crear la variable de entorno para indicar dónde está instalado el JDK y añadir a la variable PATH el directorio en el que se encuentran los binarios para poder ejecutarlos desde cualquier sitio. Para ello abrimos el archivo /etc/profile con nuestro editor favorito (como root) y añadimos las siguientes líneas al final:

```
JAVA_HOME=/usr/local/jdk1.5.0_06
PATH=$PATH:$JAVA_HOME/bin
export PATH JAVA_HOME
```

Actualizamos las variables de entorno:

```
source /etc/profile
```

Si todo ha salido bien al escribir `javac -version` deberíamos obtener el número de versión del compilador de Java. En el caso de que algo halla salido mal el sistema nos informará de que no encontró ningún ejecutable con ese nombre.

Compilar e instalar Apache

Descargamos el código fuente de la aplicación desde la web de la fundación. Descomprimimos, compilamos e instalamos:

```
cp -p httpd-2.2.0.tar.gz /usr/local/src/
cd /usr/local/src
tar xvzf httpd-2.2.0.tar.gz
rm httpd-2.2.0.tar.gz
cd /usr/local/src/httpd-2.2.0
./configure --prefix=/usr/local/apache --enable-module=most --enable-mods-
shared=most
make
make install
```

Con esto habremos instalado Apache en /usr/local/apache. Vamos a probar la configuración por defecto y a intentar iniciar el servidor:

```
/usr/local/apache/bin/apachectl configtest
/usr/local/apache/bin/apachectl start
```

Si todo ha funcionado correctamente deberíamos poder abrir la URL http://localhost en un navegador y ver la página de bienvenida de Apache.

Detengamos Apache hasta que instalemos Tomcat y el conector:

```
/usr/local/apache/bin/apachectl stop
```

Compilar e instalar Tomcat

Descargamos la aplicación desde su web. En este caso no tenemos más que descomprimir el archivo en el directorio que queramos, ya que se trata de una aplicación Java.

```
cp -p apache-tomcat-5.5.16.tar.gz /usr/local/
cd /usr/local
tar xvzf apache-tomcat-5.5.16.tar.gz
rm apache-tomcat-5.5.16.tar.gz
```

De nuevo vamos a editar /etc/profile para añadir la variable de entorno CATALINA_HOME:

```
CATALINA_HOME=/usr/local/apache-tomcat-5.5.16
export CATALINA_HOME
```

Y actualizamos:

```
source /etc/profile
```

Por último ejecutamos el script de iniciación de tomcat:

```
/usr/local/apache-tomcat-5.5.16/bin/startup.sh
```

Se debe ver la página de bienvenida de Tomcat introduciendo la URL http://localhost:8080 en un navegador.

Por ahora vamos a parar Tomcat

```
/usr/local/apache-tomcat-5.5.16/bin/shutdown.sh
```

En este momento tenemos instalados ambos servidores que correrían de forma independiente, con Apache escuchando en el puerto 80 y Tomcat escuchando en el 8080.

Instalación de Axis

1. Bajar la versión **binaria** de Axis en: http://xml.apache.org/axis .

2. Descomprimir el archivo *Tar* de Axis un directorio temporal, esto genera un directorio llamado `axis-<numero_de_version>`.

3. Dentro de este directorio existe otro sub-directorio llamado `lib`, dentro del cual residen las distintas librerías (archivos JAR) necesarias para ejecutar Axis.

Configuración de Tomcat

Dentro de la carpeta del directorio donde se descomprimio axis-<version> hay un directorio llamado webapps, y dentro de este directorio hay una carpeta llamada axis. Copia el directorio completo axis al directorio: TOMCAT_HOME/webapps/axis.

Para ver si Axis está instalado mostramos la URL *http://localhost:8080/axis/happyaxis.jsp* y nos mostrará una página de bienvenida con la configuración de Axis.

Para tener todas las librerias de axis copia los archivos .jar de la carpete de axis-<version>/lib a la carpeta donde esta el jdk de java en el directrio $JAVA_HOME/jre/lib/ext.

Ejemplo de un servicio Web con axis

Para crear un Servicio Web creamos las clases con la lógica y métodos que deseamos para nuestro Servicio Web. Para nuestro ejemplo vamos a crear un Servicio Web que proporcione métodos para sumar, restar, multiplicar y dividir dos números enteros.

Calculadora.java

```
/**
 * Servicio Web que realiza las operaciones Suma, Resta, Multiplicación y División
de dos números
 */
public class Calculadora {
        /**
         * Realiza la suma dos números enteros
         * @param x Primer operando
         * @param y Segundo operando
         * @return Devuelve el resultado de la operación (x+y)
         */
        public int suma(int x, int y) {
                return x + y;
        }

        /**
         * Realiza la resta dos números enteros
         * @param x Primer operando
         * @param y Segundo operando
         * @return Devuelve el resultado de la operación (x-y)
         */
        public int resta(int x, int y) {
                return x - y;
        }


        /**
         * Realiza la multiplicación de dos números enteros
         * @param x Primer operando
         * @param y Segundo operando
         * @return Devuelve el resultado de la operación (x*y)
         */
        public int multiplica(int x, int y) {
                return x * y;
        }

        /**
         * Realiza la división de dos números enteros
         * @param x Primer operando
         * @param y Segundo operando
         * @return Devuelve el resultado de la operación (x/y)
         */
        public int divide(int x, int y) {
                return x / y;
        }
}
```

Activación del Servicio Web

Ahora vamos a desplegarlo (= activarlo) en el Tomcat para que pueda ser invocado desde otra aplicación.

(Axis también proporciona un servidor Stand Alone para probar Servicios Web, pero para nuestro ejemplo nos apoyaremos en el servidor de aplicaciones Tomcat)

1. Renombrar el fichero Calculadora.java a Calculadora.jws y copiarlo al directorio TOMCAT_HOME/webapps/axis. (Si hemos instalado correctamente AXIS, todos los ficheros acabados en .jws serán procesados por Axis.). De esta manera la primera vez que se invoque el Web Service, será compilado automáticamente.
2. Compilar el fichero Calculadora.java y colocamos el .class en el directorio TOMCAT_HOME/webapps/axis/WEB-INF/classes. Ahora definimos su descriptor de activación o despliegue (wsdd) y ejecutamos el comando:

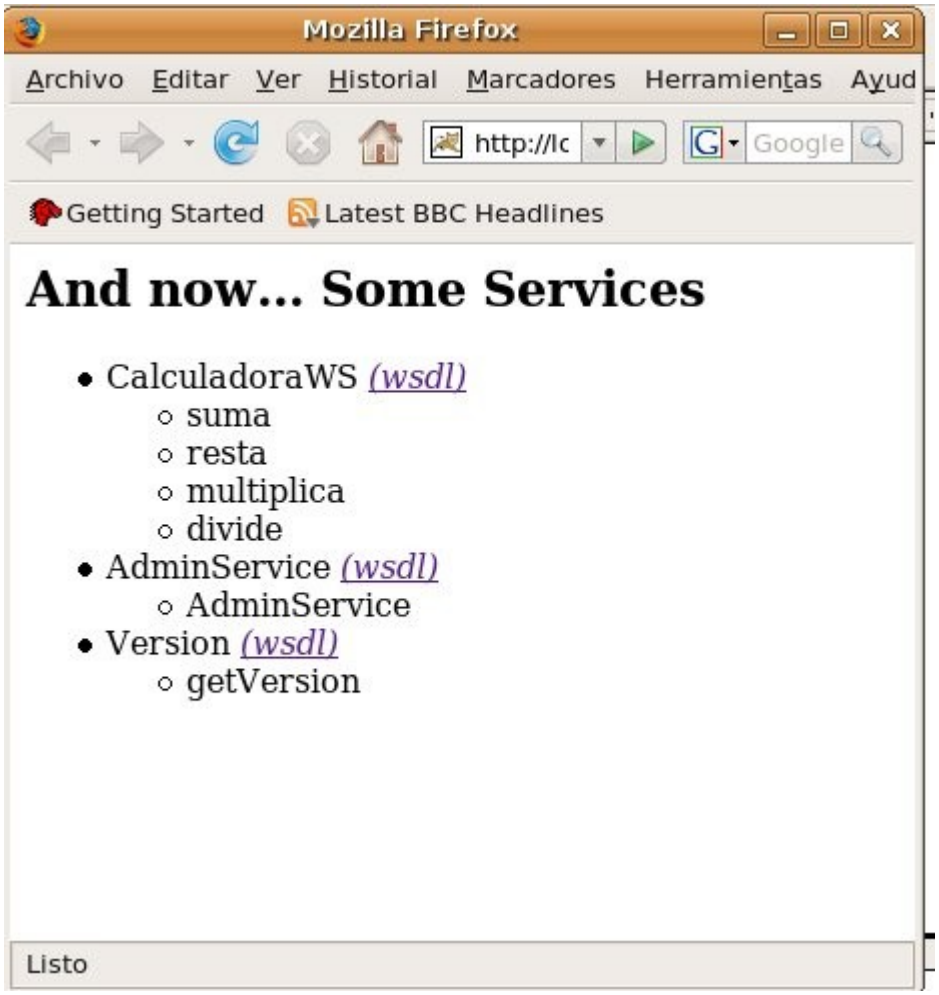
java org.apache.axis.client.AdminClient Calculadora.wsdd

Calculadora.wsdd

```
<deployment
        xmlns="http://xml.apache.org/axis/wsdd/"
        xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">

   <!--
         Definimos el servicio Web a activar:
         Nombre y tipo de Servicio Web.
         RPC Llamadas a procedimientos remotos con ejecución síncrono
   -->

   <service name="CalculadoraWS" provider="java:RPC">

        <!-- Nombre de la clase que implementa los métodos expuestos -->
       <parameter name="className" value="Calculadora"/>

        <!-- Expone todos los métodos como visibles desde el exterior -->
       <parameter name="allowedMethods" value="*"/>
   </service>
</deployment>
```

Ahora si deseamos ver una lista de los Servicios Web activados en la máquina local ejecutamos navegamos a la dirección:
http://localhost:8080/axis/servlet/AxisServlet

Deberá aparecer nuestro Servicio Web CalculadoraWS y una lista con los métodos que exporta.



Creación de un cliente Nativo

```
import org.apache.axis.client.Call;
import org.apache.axis.client.Service;
import org.apache.axis.encoding.XMLType;
import javax.xml.rpc.ParameterMode;

public class ejemplo3
{
  public static void main(String [] args) throws Exception {
     String endpoint = "http://148.247.199.145:8080/axis/Calculadora.jws";
     if (args == null || args.length != 3) {
        System.err.println("Uso: ClienteCalc operacion arg1 arg2");
        return;
     }
     String method = args[0];
     Integer i1 = new Integer(args[1]);
     Integer i2 = new Integer(args[2]);
     Service  service = new Service();
     Call     call    = (Call) service.createCall();
     call.setTargetEndpointAddress( new java.net.URL(endpoint) );
     call.setOperationName( method );
     call.addParameter( "op1", XMLType.XSD_INT, ParameterMode.IN );
     call.addParameter( "op2", XMLType.XSD_INT, ParameterMode.IN );
     call.setReturnType( XMLType.XSD_INT );
     Integer ret = (Integer) call.invoke( new Object [] { i1, i2 });
     System.out.println("Resultado : " + ret);
  }
}
```

Este Cliente que accesa el "Web-Service" es una simple Clase que puede ser ejecutada de una consola ("shell"), otra alternativa pudo haber sido diseñar otro Servlet o posiblemente un EJB que accesará la funcionalidad; para compilar este Cliente que accesa el "Web-Service" es necesario que las diversas librerías de *Axis* se encuentren en el CLASSPATH de compilación, esto se debe a que el Cliente utiliza diversas Clases *Axis* para llevar acabo la comunicación con el "Web-Service".

Una vez compilada esta Clase basta ejecutar una secuencia como la siguiente para llamar el "Web-Service":

```
#ejemplo3 suma 3 3
Resultado 6

#ejemplo3 resta 3 3
Resultado 0

#ejemplo3 divide 3 3
Resultado 1

#ejemplo3 multiplica 3 3
Resultado 9
```